

Software–Projekt 1 2013

VAK 03-BA-901.02

Projektplan

David Brinkmann	david.brinkmann@uni-bremen.de	2696099
Patrick Damrow	damsen@tzi.de	2056170
Habenicht, Arnaud	mephisto@uni-bremen.de	2260395
Daniel Pupat	s_aydi4h@uni-bremen.de	2703053
Michael Sauerweis	s_rfkrpo@uni-bremen.de	2699739
Leopold Siakeu	henrileopold@yahoo.fr	2193984

Inhaltsverzeichnis

1	Einleitung	3
1.1	Projektübersicht	3
1.1.1	Hauptarbeitsaktivitäten und –produkte	3
1.1.2	Haupt–Meilensteine und grober Zeitplan	3
1.1.3	Mitarbeiter	5
1.2	Auszuliefernde Produkte	6
1.3	Referenzen	6
2	Managementprozess	6
2.1	Managementprozess und –prioritäten	6
2.2	Annahmen, Abhängigkeiten und Einschränkungen	6
2.3	Risikomanagement	7
2.4	Projektüberwachung	10
2.5	Mitarbeiter	10
3	Arbeitspakete, Zeitplan und Budget	11
3.1	Arbeitspakete	11
3.2	Zeitplan und Abhängigkeiten	12
3.3	Ressourcenanforderung	12

Version und Änderungsgeschichte

Die aktuelle Versionsnummer des Dokumentes sollte eindeutig und gut zu identifizieren sein, hier und optimalerweise auf dem Titelblatt.

Version	Datum	Änderungen
0.1	27.04.2013	Arbeitspakete und Abhängigkeiten
0.2	27.04.2013	Risikomanagement
0.3	27.04.2013	Managementprozess 2.2
0.4	27.04.2013	Mitarbeiter hinzugefügt
1.0	27.04.2013	Erste veröffentlichte Version.
1.1	08.05.2013	Korrektur nach Abgabe

1 Einleitung

Erstellt von: David Brinkmann und Patrick Damrow

1.1 Projektübersicht

In SWP-1 ist ein Bibliothekssystem zu erstellen. Dieses System beinhaltet sowohl eine Website als auch eine Android App. Die Website richtet sich an Leser (Bibliotheksbenuer) und Bibliothekare. Die App ist für die Leser gedacht. Bibliothekare können Bücher und Benutzer verwalten (anlegen und einsehen). Leser können Bücher einsehen und bewerten. Dieses System wird vermutlich in SWP-2 weiterentwickelt mit weiteren Anforderungen. Daher ist es ratsam, bereits jetzt an mögliche zukünftige Änderungen zu denken. (Quelle: <http://www.informatik.uni-bremen.de/st/Lehre/swp/mindestanforderungen.html>)

1.1.1 Hauptarbeitsaktivitäten und –produkte

Implementierung einer Bibliotheksverwaltungssoftware in Android.

1.1.2 Haupt–Meilensteine und grober Zeitplan

M1 - 02.05.13 Projektplan abgeben

Jedes Mitglied muss seinen Teil fertig gestellt haben. Anschließend wurden alle Einzelteile zusammengeführt und von allen auf Korrektheit geprüft.

M2 - 14.05.13 Anforderungsspezifikationen (Intern)

Jedes Mitglied hat seinen Teil der Anforderungsspezifikation fertiggestellt. Anschließend wurden die Teile zusammen geführt und von allen auf Korrektheit geprüft.

M3 - 16.05.13 Abgabe der Anforderungsspezifikation

Meilenstein 2 muss bereits fertig sein. Abgabe der Anforderungsspezifikation via MEMS.

M4 - 24.06.13 Architektur, Testplan, Tests (Intern)

Jedes Mitglied muss seine Aufgaben erfüllt haben. Teile wurden zusammengeführt und kontrolliert. Tests müssen implementiert sein.

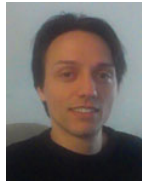
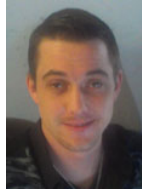

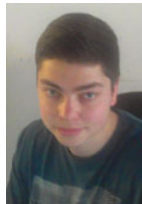
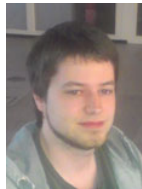

M5 - 27.06.13 Architekturbeschreibung, Testplan und Schnittstellentests fertig

Meilenstein 4 muss bereits erreicht worden sein. Tests wurden lauffähig implementiert. Abgabe via MEMS.

M6 - 19.07.13 Vollständige Abgabe des Dokuments und Software

Implementierung in der Woche zum 19.07. Vollständige Abgabe der Dokumente und der lauffähigen Software.

1.1.3 Mitarbeiter

Name	Email	
Brinkmann, David	david.brinkmann@uni-bremen.de	
Damrow, Patrick	damsen@tzi.de	
Habenicht, Arnaud	mephisto@informatik.uni-bremen.de	
Pupat, Daniel	s_aydi4h@uni-bremen.de	
Sauerwein, Michael	s.rfkrpo@uni-bremen.de	
Siakeu, Leopold	henrileopold@yahoo.fr	

1.2 Auszuliefernde Produkte

1.3 Referenzen

2 Managementprozess

Erstellt von: Leopold Siakeu und Arnaud Habenicht

2.1 Managementprozess und –prioritäten

Erstellt von: Patrick Damrow

Die Prioritäten liegen in der Fertigstellung der vom Kunden gewünschten Anwendung zur Verwaltung einer Bibliothek bzw. die Implementierung dieser Anwendung auf Android-fähigen Endbenutzergeräten. Die Zufriedenheit des Kunden steht hier im Vordergrund. Um dies zu gewährleisten, werden wir einen Qualitätsmanager bestimmen, der regelmäßig überprüfen soll, ob alle Anforderungen erfüllt sind. Darüber hinaus trifft sich die Gruppe einmal wöchentlich Mittwochs für 2 Stunden von 12 - 14 Uhr an der Uni, um über die Qualität, den Fortschritt und auftretende Probleme zu sprechen. Zur Qualitätsgewährleistung der Dokumente treffen sich alle Mitglieder mindestens zwei Tage vor Abgabe des eweiligen Dokuments, um einen Zeitpuffer für Verbesserungen zu haben.

Für ein gutes Gruppenklima, welches zwingend notwendig für ein funktionierendes und produktives Team ist, planen wir ausseruniversitäre Aktivitäten. Dies soll für Vertrauen und Zusammengehörigkeit sorgen. Intern haben wir SSpielregelnneingeführt um Unzuverlässigkeit zu unterbinden und somit das Vertrauen zusätzlich zu stärken. Die Kommunikation unserer Gruppe wird meist elektronisch über E-Mail von statten gehen. Teilweise kommunizieren wir über Telefon und/oder WhatsApp, sowie den wöchentlichen Meetings.

2.2 Annahmen, Abhängigkeiten und Einschränkungen

Erstellt von: Patrick Damrow

Das Projekt beruht auf der Annahme, dass dem Kunden und Mitarbeitern mit dieser Software die Verwaltung der Bibliothek vereinfacht wird. Diese muss zweisprachig gestaltet sein. Darüberhinaus nehmen wir an, dass jedes Gruppenmitglied solide Java Grundkenntnisse hat, da alle Mitglieder der Gruppe mindestens Praktische Informatik 1 an der Universität Bremen bestanden haben.

Die meisten Gruppenmitglieder besitzen ein Smartphone oder Tablet mit einem Android Betriebssystem, was das Testen der Software erleichtert, da es direkt am Endgerät getestet werden kann. Falls diese Ausfallen sollten, benutzen wir den Emulator des

AndroidSDK. Das Problem der Abhängigkeit an einen festen Ort gebunden zu sein, lösen wir indem jedes Gruppenmitglied ein eigenes Notebook besitzt. Bei Ausfall können wir auf Rechner im Rechnerpool der Ebene 0 im MZH an der Uni Bremen zurückgreifen.

Als zusätzliche Abhängigkeit wären noch die Gesundheit der Mitglieder, das SVN (Subversion Repository) und LaTeX zu nennen. Unser Arbeitsstundenkontingent ist stark von der Gesundheit unserer Mitglieder abhängig, da ein langfristiger Ausfall viel mehr Arbeit für die anderen Mitglieder bedeuten würde. Das SVN ist wichtig, da wir sonst nicht die Arbeiten von 6 Personen an einem Dokument bewältigen könnten und LaTeX wird von uns als Textsatz-Software zur Erstellung unserer Dokumente verwendet.

Zeitliche Einschränkungen entsche zusätzlich dadurch, dass einige Mitglieder Regelmäßig arbeiten gehen und andere universitäre Veranstaltungen besuchen.

Eine Einschränkung besteht in der Platzformabhängigkeit von Android, sowie der verfügbaren Zeit zur Fertigstellung dieses Projekts. Zudem ist die erfolgreiche Fertigstellung des Projekts Voraussetzung zum Bestehen des Moduls SWP-1.

2.3 Risikomanagement

“Projekte, Prozesse und Produkte (IT-Infrastrukturen, Software) im IT-Bereich unterliegen – das steht außer Zweifel – immer einem gewissen Risiko. Eine Vielzahl von Einflussfaktoren kann dazu führen, dass das Erreichen der angestrebten Ziele gefährdet oder gar erhebliche negative Folgewirkungen denkbar sind (etwa aus Fehlern oder Verzögerungen in der Ausführung der IT-Prozesse oder der IT-Projekte).“ (Quelle: Tiemeyer (Hrsg.), 2011, Handbuch IT-Management, 4. Auflage)

Es wird versucht, mögliche Risiken im Voraus zu erkennen und wenn möglich, Maßnahmen zu ergreifen, welche entweder die Eintrittswahrscheinlichkeit und/oder die Schadenhöhe eindämmen. Des Weiteren wird während des Projekts stets auf neu aufkommende Risiken geachtet und auf dadurch entstehende Probleme frühzeitig reagiert. Die Kontrolle über Risiken während des Projekts liegt in der Obhut des Phasenleiters, welcher gegebenenfalls Gegenmaßnahmen einleitet, Gruppentreffen organisiert oder in Arbeitsschritte interveniert.

Absehbare Risiken sind in der nachfolgenden Tabelle aufgeführt. Gegenmaßnahmen werden im Anschluss an diese erläutert.

Risiko	Eintrittswahrsch. (1-3)	Schadenshöhe (1-6)	Risikohöhe
Unzuverlässigkeit	2	3	6
Personalausfall	1	2	2
Inkompetenz	1	4	4
Technik	1	5	5
Kommunikation	2	3	6
Zeitmangel	3	4	12
Unstimmigkeiten	2	4	8
Mangelnde Motivation	3	3	9

(1 niedrigste, 3/6 höchste)

Maßnahmen gegen:

Unzuverlässigkeit: Regeln aufstellen, Konsequenzen (Abmahnungen), Ausschluss aus Gruppe

EW: 1 SH: 3

- Präventive Maßnahmen: Es wird von Anfang über Regeln besprochen und Regeln festgelegt. Alle müssen von getroffenen Regeln Bescheid wissen, denn deren Missbrauch Konsequenzen hat, wie zum Beispiel: Abmahnungen.
- Reaktive Maßnahmen: Sollte ein oder mehrerer Mitglieder der Gruppe sich unvernünftig benehmen (Treffen ignorieren, Anforderungen nicht anhalten, u.s.w.) können die anderen Mitglieder ihn verwarnen. Falls sein Verhalten sich nicht verbessert, scheidet das Mitglied (nach Absprache der Gruppe, mit einem Tutor bzw. mit dem Professor) aus der Gruppe aus.

Personalausfall: Aufgaben neu verteilen, bei Ausschluss eines Mitgliedes Projektanforderungen verringern (Absprache mit Tutor)

EW: 1 SH: 1

- Präventive Maßnahmen: (kann nicht vorgebeugt werden).
- Reaktive Maßnahmen: Sollte ein oder mehrerer Mitglieder an einer Krankheit bzw. unter Unfallfolgen leiden (Erkältung, Knochenbruch, ...), werden weitere Mitglieder seine Aufgabe(n) vorläufig übernehmen. Eine Verteilung wird in der Gruppe besprochen.

Inkompetenz: Gegenseitige Hilfe, Tauschen der Arbeiten, Wissen aneignen

EW: 1 SH: 4

- Präventive Maßnahmen: Eine Umfrage wird durchgeführt um herauszufinden, wer was am besten kann. Dadurch kann man eine optimale Aufgabeverteilung planen.
- Reaktive Maßnahmen: Sollte ein oder mehrerer Mitglieder die erwarteten Grundkenntnisse nicht vorweisen, müssen sie frühzeitig Bescheid geben, so

dass sie von den anderen Team-Mitglieder unterstützt werden können, je nachdem wie groß die Notwendigkeit dazu ist, kann eine neue Verteilung geplant werden.

Technik: Ständige Backups, Ausweichen auf Uni-Rechner, Ausweichen auf alternative Software

EW: 1 SH: 3

- Präventive Maßnahmen: Alle Mitglieder müssen dafür sorgen, dass regelmäßige Backups durchgeführt werden. Sie müssen ebenfalls zur Sicherung der Arbeit eine Kopie auf einer beliebigen Datenträger speichern.
- Reaktive Maßnahmen: Sollte ein Mitglied (von Zuhause aus, auf seinem Notebook) keinen Zugang zum Server haben, stehen ihm noch die Uni-Rechner zu Verfügung.

Kommunikation: Regelmäßige Treffen, Zusätzlicher Austausch über Mails/Telefon
EW:1 SH: 3

- Präventive Maßnahmen: Regelmäßige Treffen müssen geplant werden, nicht nur im Rahmen der Arbeitsbesprechung, sondern auch um die gesellschaftliche bzw. die soziologische Ebene unter Mitglieder zu pflegen.
- Reaktive Maßnahmen: Sollte es doch vorkommen, dass ein oder mehrerer Mitglieder aus irgendeinem Grund nicht zu Treffen erscheinen können, bekommen die Abwesenden per Mail/Telefon ein Feedback.

Zeitmangel: Vorhandene Zeit organisieren, Ausweichen auf Wochenenden, Kleine Zeitpuffer einplanen

EW: 2 SH: 4

- Präventive Maßnahmen: Um zu vermeiden, dass es der Gruppe an Zeit fehlt, wird mehr Zeit als nötig für eine Abgabe geplant, so dass bei Umständen noch mehr Zeit in Anspruch genommen wird, ohne dass es die Arbeit des Teams direkt beeinflusst bzw. sofort hindern.
- Reaktive Maßnahmen: Beim Zeitmangel, werden zusätzliche Termine geplant (Überstunden am Wochenende sind nicht ausgeschlossen). Im schlimmsten Fall kann sich die für eine Arbeitspaket geplante Anzahl von Mitgliedern verdoppelt, um an Zeit wahrscheinlich aufzuholen.

Unstimmigkeiten: Diskussion in der Gruppe, neutrale Person entscheiden lassen

EW: 1 SH: 3

- Präventive Maßnahmen: Jede Mitglieder sollten auf sachlicher Ebene handeln. Das heißt nicht andere Ideen aus Gefühl (wie Stolz, Schuldgefühl,

Freundschaft) ablehnen bzw. annehmen, sondern aus praktischer Sicht betrachten.

- *Reaktive Maßnahmen:* Sollten sich Gruppenmitglieder über eine Idee nicht einigen können, entscheidet in einer Gruppenbesprechung die Mehrheit, welche Option die beste wäre (es wird davon ausgegangen, dass die vorgeschlagenen Ideen umsetzbar sind). Sollte in der Gruppe keine Entscheidung getroffen werden, schließt eine neutrale Person, die sich auskennt, die Debatte ab.

Motivation: Zeiten einplanen, sich mit der Gruppe zu treffen, ohne dabei am Projekt zu arbeiten.

EW: 2 SH: 3

- *Präventive Maßnahmen:* Eine positive Einstellung ist in der Gruppe empfohlen. Gegenseitige Rücksicht ist von Vorteil, so dass man diejenigen, den es an Kraft fehlt, ermuntern kann.
- *Reaktive Maßnahmen:* Sollte es der Gruppe an Begeisterung und Motivation fehlen, werden Treffen “aufs Bier” organisiert, so dass die Mitglieder für eine Zeit lang abschalten können und dadurch sich gegenseitig wieder stärken.

2.4 Projektüberwachung

Der Phasenleiter organisiert regelmäßige Treffen und informiert sich über den aktuellen Stand der einzelnen Mitarbeiter. Diesen Stand vergleicht er mit dem aufgestellten Zeitplan und ergreift nötigenfalls Maßnahmen zur Beschleunigung der noch anstehenden Arbeiten. Die Kommunikation unter den Mitarbeitern stellt sicher, dass etwaige Defizite frühzeitig erkannt werden und gegenseitige Unterstützung gewährleistet wird, zur Einhaltung der vorgegebenen Zeiten.

2.5 Mitarbeiter

Vorausgesetzt werden solide Grundkenntnisse in Java, soweit diese im Modul PI-1 vermittelt wurden. Des Weiteren wird vorausgesetzt, dass sich jeder Mitarbeiter eigenständig in neue Themenbereiche einarbeiten kann. Außerdem sollte jeder Mitarbeiter in der Lage sein eigene Defizite zu erkennen und diese rechtzeitig der Gruppe mitteilen können.

Für die Teilbereiche (wie. z.B. Layout) sollte der Mitarbeiter über die entsprechenden Kenntnisse verfügen.

3 Arbeitspakete, Zeitplan und Budget

Erstellt von: Michael Sauerwein und Daniel Papat

3.1 Arbeitspakete

1 Initialer Projektplan In dem Arbeitspaket wird der Projektplan erstellt, mit einer Einleitung, dem Managementprozess und den Arbeitspaketen und einen Zeitplan. Phasenleiter dieses Projektes ist David Brinkmann, wobei alle zusammen daran arbeiten, da nur eine Aufgabe ist, in diesem Zeitraum zu erledigen ist. Dieses Projekt soll im Zeitraum vom 24.04 bis zum 2.05 andauern.

2 Anwendungsspezifikation Im Arbeitspaket Anwendungsspezifikation wird eine detaillierte Darstellung des Datenmodells, der Anwendungsfälle und der Aktionen geben. Phasenleiter dieses Pakets ist Patrick Damrow, wobei wieder alle Gruppenmitglieder an dem Projekt arbeiten. Dieses Arbeitspaket soll im Zeitraum vom 2.05 bis zum 16.05 fertiggestellt werden

3 Architekturbeschreibung, Testplan und Schnelltests In dem Arbeitspaket wird die Architektur der Implementierungen erstellt, sowie schon vorläufige Testfälle. Phasenleiter dieses Projektes ist Michael Sauerwein und es werden wieder alle Mitglieder an diesem Projekt arbeiten. Dieses Arbeitspaket wird vom 16.05 bis zum 27.6 fertiggestellt werden

4.0 Arbeitspakete 4.1-4.4 werden alle separat in der Implementierungsphase vom 15.07 bis zum 19.07 ablaufen. Der Phasenleiter der Implementierungsphase ist Daniel Papat, der einen Überblick über die einzelnen Pakete haben soll.

4.1 Server: Der Server soll zur Kommunikation und Datenaustausch mit dem Client dienen, sowie einen Zugriff auf die Datenbank haben. Der Server wird von David Brinkmann bearbeitet. Dieser Vorgang soll vom 15.07 bis zum 18.07 andauern.

4.2 Client: Der Client soll mit dem Server kommunizieren, in dem er Daten vom Server abgerufen bzw. mit dem Server austauscht. Der Client wird von Brice Arnaud Habenicht bearbeitet. Dieser Vorgang soll vom 15.07 bis zum 18.07 andauern.

4.3 GUI: Die GUI soll übersichtlich und Nutzerfreundlich sein. Die GUI wird von Patrick Damrow(50 %) und Michael Sauerwein(50 %) bearbeitet. Dieser Vorgang soll vom 15.07 bis zum 18.07 andauern.

4.4 Datenbank:

- Es soll eine Datenbank von Büchern erstellt werden, die in der Bibliothek vorhanden bzw. ausgeliehen werden können.
- Es soll eine Datenbank von Nutzern erstellt werden, in welcher Kunden der Bibliothek enthalten sind. Die Datenbanken werden von Daniel Papat bearbeitet. Dieser Vorgang soll vom 15.07 bis zum 18.07 andauern.

5 Tests: In diesem Arbeitspaket sollen Tests geschrieben werden, um den Code auf Fehler zu prüfen und so aufkommende Probleme vorzubeugen und überprüfen, ob das Programm überhaupt funktioniert. Die Tests werden von Daniel Papat, Michael Sauerwein und Leopold Siakeu bearbeitet, wobei Leopold der Phasenleiter ist. Die Tests dauern vom 18.07 bis zu 19.07. Die Software Implementierung muss vorher abgeschlossen sein.

6.1 Dokumentation: Bei der Dokumentation soll ein Handbuch erstellt werden, welches zur Beschreibung und Benutzung für Mitarbeiter und Nutzer der Bibliothek dienen soll. Die Dokumentation wird von David Brinkmann bearbeitet. Die Dokumentation soll im Zeitraum vom 18.07 bis zum 19.07 erstellt werden. Die Software Implementierung muss vorher abgeschlossen sein.

6.2 Skript: Bei dem Arbeitspaket soll ein Installationsskript, zur Installation des Programms auf einem Computer oder Android fähigen Geräten, erstellt werden. Das Skript wird von Brice Arnaud Habenicht angefertigt. Das Skript soll im Zeitraum vom 18.07 bis zum 19.07 angefertigt werden. Die Software Implementierung muss vorher abgeschlossen sein.

3.2 Zeitplan und Abhängigkeiten

Als jeweilige Meilensteine wählen wir die auf der Website angegebenen Abgabetermine. Bevor die Architekturbeschreibung, Schnittstellentests und der Testplans erstellt werden können, muss die Anwendungsspezifikation ausgearbeitet sein. Die gesamte Planungsphase muss spätestens am 06.07.2013 abgeschlossen sein. In der Woche des Blockkurses müssen zuerst die Implementierungen abgeschlossen sein, bevor die Tests und das Installationsskript bearbeitet werden können.

3.3 Ressourcenanforderung

Bei den Planungsarbeiten sind alle Mitglieder der Gruppe beteiligt. Während des Blockkurses werden die Ressourcen gemäß des Diagrammes aufgeteilt. In der letzten, kritischen Phase arbeiten zwei Gruppen zu je drei Leuten.

Abbildung 1: Gantt1
















		Vorgang	Vorgangsname	Dauer	Anfang	Ende	Vorgänger	Ressourcennam
1			Initialer Projektplan	6,5 Tage	Mit 24.04.13	Don 02.05.13		Brinkmann; Damrow;Habenicht
2			Abgabe initialer Projektplan	0 Tage	Don 02.05.13	Don 02.05.13	1	
3			Anforderungsspezif	10,5 Tage?	Don 02.05.13	Don 16.05.13	2	Brinkmann; Damrow;Habenicht
7			Architekturbeschreibung Testplan und Schnittstellentests	30 Tage	Fre 17.05.13	Don 27.06.13	3	Brinkmann; Damrow; Habenicht;Pupat
11			Diverses	11,5 Tage	Fre 28.06.13	Mon 15.07.13	7	
17			Server implementieren	3,5 Tage	Mon 15.07.13	Don 18.07.13	11	Brinkmann
18			Client implementieren	3,5 Tage	Mon 15.07.13	Don 18.07.13	11	Habenicht
19			GUI	3,5 Tage	Mon 15.07.13	Don 18.07.13	11	Damrow[50%];S;
20			Datenbank	3,5 Tage	Mon 15.07.13	Don 18.07.13	11	Pupat
21			Skript	1 Tag	Fre 19.07.13	Fre 19.07.13	17;18;19;20	Damrow[50%];H
22			Dokumentation	1 Tag	Fre 19.07.13	Fre 19.07.13	17;18;19;20	Brinkmann
24			Test	1 Tag	Fre 19.07.13	Fre 19.07.13	17;18;19;20	Pupat[50%];Sauk
25			Vollständige Abgabe	0 Tage	Fre 19.07.13	Fre 19.07.13	24	

Abbildung 2: Gantt2

